
UUU UUU EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT PPPPPPPPPPPP SSSSSSSSSSS YYY YYY
UUU UUU EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT PPPPPPPPPPPP SSSSSSSSSSS YYY YYY
UUU UUU EEEEEEEEEEEEEEE TTTTTTTTTTTTTTT PPPPPPPPPPPP SSSSSSSSSSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEE TTT PPP PPP SSS YYY YYY
UUU UUU EEEEEEEEEEE TTT PPPPPPPPPPPP SSSSSSSSS YYY
UUU UUU EEEEEEEEEEE TTT PPPPPPPPPPPP SSSSSSSSS YYY
UUU UUU EEEEEEEEEEE TTT PPPPPPPPPPPP SSSSSSSSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUU UUU EEE TTT PPP SSS YYY
UUUUUUUUUUUUUUU EEEEEEEEEEEEEEE TTT PPP SSSSSSSSSSS YYY
UUUUUUUUUUUUUUU EEEEEEEEEEEEEEE TTT PPP SSSSSSSSSSS YYY
UUUUUUUUUUUUUUU EEEEEEEEEEEEEEE TTT PPP SSSSSSSSSSS YYY
UUUUUUUUUUUUUUU EEEEEEEEEEEEEEE TTT PPP SSSSSSSSSSS YYY

```
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS    SSSSSSSS  5555555555    222222
  SSSSSSSS    AAAAAA    TTTTTTTTTT    SSSSSSSS    SSSSSSSS    SSSSSSSS  5555555555    222222
SS            AA  AA        TT      SS          SS          SS            55                22
SS            AA  AA        TT      SS          SS          SS            55                22
SS            AA  AA        TT      SS          SS          SS            555555            22
SS            AA  AA        TT      SS          SS          SS            555555            22
  SSSSSS      AA  AA        TT        SSSSSS      SSSSSS      SSSSSS              55            22
  SSSSSS      AA  AA        TT        SSSSSS      SSSSSS      SSSSSS              55            22
        SS    AAAAAAAAAA    TT              SS          SS          SS                55      22
        SS    AAAAAAAAAA    TT              SS          SS          SS                55      22
        SS    AA  AA        TT              SS          SS          SS  55            55    22              ....
        SS    AA  AA        TT              SS          SS          SS  55            55    22              ....
SSSSSSSS      AA  AA        TT      SSSSSSSS    SSSSSSSS    SSSSSSSS      555555      2222222222        ....
SSSSSSSS      AA  AA        TT      SSSSSSSS    SSSSSSSS    SSSSSSSS        555555    2222222222        ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
0000     1          .TITLE  SATSSS52 SATS SYSTEM SERVICE TESTS  $DLCEFC (SUCC S.C.)
0000     2          .IDENT  'V04-000'
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28
0000    29  ;++
0000    30  ; FACILITY:     SYSTST (SATS SYSTEM SERVICE TESTS)
0000    31  ;
0000    32  ; ABSTRACT:
0000    33  ;
0000    34  ;                THIS MODULE CONTAINS SUBROUTINES WHICH, WHEN LINKED
0000    35  ; WITH SUCCOMMON.OBJ, FORM TEST MODULE SATSSS52 TO TEST SUCCESSFUL
0000    36  ; OPERATION OF THE $DLCEFC SYSTEM SERVICE. THE SERVICE IS INVOKED
0000    37  ; UNDER VARIOUS INPUT CONDITIONS WITH VARYING INPUT PARAMETERS. ONLY
0000    38  ; SUCCESSFUL STATUS CODES ARE EXPECTED IN THIS TEST MODULE.  CORRECT
0000    39  ; OPERATION OF THE SERVICE FOR EACH OF ITS ISSUANCES IS VERIFIED BY
0000    40  ; CHECKING FOR AN SS$_NORMAL STATUS CODE, EXPECTED RETURN ARGUMENTS
0000    41  ; AND EXPECTED FUNCTIONALITY PERFORMED.
0000    42  ;
0000    43  ; ENVIRONMENT:  USER MODE IMAGE; NEEDS CMKRNL PRIVILEGE,
0000    44  ;                DYNAMICALLY ACQUIRES OTHER PRIVILEGES, AS NEEDED.
0000    45  ;
0000    46  ; AUTHOR: THOMAS L. CAFARELLA,          CREATION DATE: JAN, 1978
0000    47  ;
0000    48  ; MODIFIED BY:
0000    49  ;
0000    50  ;                VERSION 1.5 : 25-MAY-79
0000    51  ;
0000    52  ; 01  LDJ  10/11/79     Fixed bug caused by DIB$K_LENGTH change ACG052.RNO mem
0000    53  ;
0000    54  ;--
```

```
0000     56          .SBTTL  DECLARATIONS
0000     57 ;
0000     58 ; INCLUDE FILES:
0000     59 ;
0000     60          $PRVDEF                         ; PRIVILEGE BIT DEFINITIONS
0000     61          $PHDDEF                         ; PROCESS HEADER OFFSETS
0000     62          $PQLDEF                         ; PROCESS QUOTA CODES
0000     63          $DIBDEF                         ; DEVICE INFO BLOCK OFFSETS
0000     64 ;
0000     65 ; MACROS:
0000     66 ;
0000     67 ;
0000     68 ; EQUATED SYMBOLS:
0000     69 ;
0000     70 ;
0000     71 ; OWN STORAGE:
0000     72 ;
```

```
00000000     74          .PSECT  RODATA,RD,NOWRT,NOEXE,LONG
    0000     75 TEST_MOD_NAME:: STRING  C,<SATSSS52>     ; TEST MODULE NAME
    0009     76 TEST_MOD_NAME_D: STRING I,<SATSSS52>     ; TEST MODULE NAME DESCRIPTOR
    0019     77 MSG1_INP_CTL:   STRING  I,<  SSDEF!4ZW: CONDITIONS:>
    0039     78                                          ; FAO CTL STRING FOR MSG1 IN SUCCOMMON.MAR
    0039     79 MSG3_ERR_CTL::  STRING  I,< *SSDEF!4ZW: !AS>
    0051     80                                          ; FAO CTL STRING FOR MSG3 IN SUCCOMMON.MAR
    0051     81 CREPRN:         STRING  I,<SATSSS52_CRE> ; CREATED PROCESS NAME
    0065     82 CLUS_NAME:      STRING  I,<SATSSS52_CLUS> ; SUBJECT CLUSTER NAME
    007A     83 IMAGNAM:        STRING  I,<SYSTST$RES:SATSUT10.EXE> ; IMAGE NAME FOR CREATED PROC
    0099     84 QUOTALIST:      $QUOTA  CPULM,0          ; INFINITE CPU
    009E     85                 $QUOTA  BYTLM,512        ; BYTE LIMIT FOR BUFFERED I/O
    00A3     86                 $QUOTA  FILLM,2          ; OPEN FILE COUNT LIMIT
    00A8     87                 $QUOTA  PGFLQUOTA,10     ; PAGING FILE QUOTA
    00AD     88                 $QUOTA  PRCLM,2          ; SUBPROCESS QUOTA
    00B2     89                 $QUOTA  TQELM,3          ; TIMER QUEUE ENTRY QUOTA
    00B7     90                 $QUOTA  LISTEND          ; DEFINES END OF LIST
```

```
                00000000     92          .PSECT  RWDATA,RD,WRT,NOEXE,LONG
00000008  0000  00000008     93 PRIVMASK:       .BLKQ   1               ; ADDR OF PRIVILEGE MASK (IN PHD)
0000000C  0008  0000000C     94 MBXCHAN:        .BLKL   1               ; CHAN NO. FOR MAILBOX FOR CREATED PROCESS
          000C  00000010     95 MBXCHANINFO:                            ; CHANNEL INFO RETURNED BY GETCHN
00000074  000C  00000074     96                 .LONG   DIB$K_LENGTH
00000014' 0010  00000014'    97                 .ADDRESS .+4
00000088  0014  00000088     98                 .BLKB   DIB$K_LENGTH
0000008C  0088  0000008C     99 MBXUNIT:        .BLKL   1               ; SAVE AREA FOR MAILBOX UNIT NUMBER
          008C  00000090    100 MBXBUFF:        STRING  0,120           ; MAILBOX BUFFER FOR CREATED PROCESS
00000110  010C  00000110    101 CLUS_MASK:      .BLKL   1               ; CLUSTER MASK; USED TO SET SUBJECT CLUSTER
00000114  0110  00000114    102 CLUS_STATE:     .BLKL   1               ; STATE OF SUBJECT CLUSTER
00000118  0114  00000118    103 EFN_REFCT1:     .BLKL   1               ; SAVE AREA FOR EFN WHEN REF CT = 1
```

```
                          0118   105          .SBTTL  CONDITION TABLES
                          0118   106 ;
                          0118   107 ;       ***** CONDITION TABLES FOR DLCEFC SYSTEM SERVICE *****
                          0118   108 ;
                          0118   109          COND    1,NOTARG,<CLUSTER'S PERM/TEMP SETTING AT TIME OF DLCEFC>,-
                          0118   110                    <TEMPORARY>,-
                          0118   111                    <PERMANENT>,-
                          0118   112
00000001 00000000         0163   113                       .LONG       0,1     ; TEMPORARY/PERMANENT
                          016B   114 ;
                          016B   115          COND    2,NOTARG,<CLUSTER'S REFERENCE COUNT AT TIME OF DLCEFC>,-
                          016B   116                    <ZERO>,-
                          016B   117                    <ONE (E.F. GROUP 2)>,-
                          016B   118                    <ONE (E.F. GROUP 3)>,-
                          016B   119                    <TWO (BOTH E.F. GROUPS)>,-
                          016B   120                    <FOUR (BOTH E.F. GROUPS IN TWO PROCESSES)>,-
                          016B   121
                          0217   122          COND    3,NULL
                          0218   123
                          0218   124          COND    4,NULL
                          0219   125
                          0219   126          COND    5,NULL
                          021A   127
                 00000000   128          .PSECT  SATSSS52,RD,WRT,EXE
```

```
                                                I 16
0000    130          .SBTTL  TM_SETUP, TM_CLEANUP
0000    131 ;++
0000    132 ; FUNCTIONAL DESCRIPTION:
0000    133 ;
0000    134 ;               TM_SETUP AND TM_CLEANUP ARE CALLED TO PERFORM
0000    135 ; REQUIRED HOUSEKEEPING AT THE BEGINNING AND END, RESPECTIVELY, OF
0000    136 ; TEST MODULE EXECUTION.
0000    137 ;
0000    138 ; CALLING SEQUENCE:
0000    139 ;
0000    140 ;       BSBW TM_SETUP   BSBW TM_CLEANUP
0000    141 ;
0000    142 ; INPUT PARAMETERS:
0000    143 ;
0000    144 ;       NONE
0000    145 ;
0000    146 ; IMPLICIT INPUTS:
0000    147 ;
0000    148 ;       NONE
0000    149 ;
0000    150 ; OUTPUT PARAMETERS:
0000    151 ;
0000    152 ;       NONE
0000    153 ;
0000    154 ; IMPLICIT OUTPUTS:
0000    155 ;
0000    156 ;       TM_SETUP:  COND TABLE INDEX REGISTERS (R2,3,4,5,6) CLEARED;
0000    157 ;                  ALL PRIVILEGES ACQUIRED.
0000    158 ;
0000    159 ; COMPLETION CODES:
0000    160 ;
0000    161 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0000    162 ;
0000    163 ; SIDE EFFECTS:
0000    164 ;
0000    165 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0000    166 ;       (VIA RSB) IF ERROR ENCOUNTERED.
0000    167 ;
0000    168 ;--
0000    169
0000    170
0000    171
0000    172 TM_SETUP::
                52   D4  0000    173          CLRL    R2                          ; INITIALIZE
                53   D4  0002    174          CLRL    R3                          ; .. CONDITION
                54   D4  0004    175          CLRL    R4                          ; .... TABLE
                55   D4  0006    176          CLRL    R5                          ; ...... INDEX
                56   D4  0008    177          CLRL    R6                          ; ........ REGISTERS
            FFF3'    30  000A    178          BSBW    MOD_MSG_PRINT               ; PRINT TEST MODULE BEGIN MSG
00000000'EF 00000000'EF  DE  000D  179          MOVAL   TEST_MOD_SUCC,TMD_ADDR  ; ASSUME END MSG WILL SHOW SUCCESS
   03   00 00000000'8F   F0  0018  180          INSV    #SUCCESS,#0,#3,MOD_MSG_CODE ; ADJUST STATUS CODE FOR SUCCESS
           00000000'EF          0020
                        0025   181          MODE    TO,S$,KRNL                  ; KERNEL MODE TO ACCESS PHD
        59 00000000'9F   D0  0048  182          MOVL    @#CTL$GL_PHD,R9             ; GET PROCESS HEADER ADDRESS
        00000000'EF   69 DE  004F  183          MOVAL   PHD$Q_PRIVMSK(R9),PRIVMASK ; GET PRIV MASK ADDRESS
                        0056   184          MODE    FROM,S$ ; BACK TO USER MODE
                        0057   185          PRIV    ADD,ALL                     ; GET ALL PRIVILEGES
```

```
                                                    J 16
                     0077   186          $SETPRN_S TEST_MOD_NAME_D       ; SET PROCESS NAME
                     0084   187          SS_CHECK  NORMAL                ; CHECK STATUS CODE RETURNED FROM SETPRN
                     00B2   188          $CREMBX_S CHAN=MBXCHAN, LOGNAM=CREPRN, - ; GET MAILBOX FOR PROCESS
                     00B2   189                    MAXMSG=#120, PROMSK=#0, BUFQUO=#240
                     00D7   190          SS_CHECK NORMAL                 ; CHECK NORMAL COMPLETION
                     0105   191          $GETCHN_S CHAN=MBXCHAN, -       ; GET CHAN INFO (UNIT NUMBER)
                     0105   192                    PRIBUF=MBXCHANINFO
                     011F   193          SS_CHECK NORMAL                 ; CHECK NORMAL COMPLETION
00000088'EF  00000020'EF  3C 014D   194          MOVZWL  MBXCHANINFO+8+DIB$W_UNIT,MBXUNIT ; SAVE MAILBOX UNIT NUMBER
                          05 0158   195          RSB                             ; RETURN TO MAIN ROUTINE
                     0159   196 TM_CLEANUP::
                     0159   197          $DELMBX_S  MBXCHAN              ; DELETE TERMINATION MAILBOX
                FE96' 30 0167   198          BSBW    MOD_MSG_PRINT           ; PRINT TEST MODULE END MSG
                          05 016A   199          RSB                             ; RETURN TO MAIN ROUTINE
```

```
                        016B   201          .SBTTL  CONDITION SUBROUTINES - SETUP AND CLEANUP
                        016B   202 ;++
                        016B   203 ; FUNCTIONAL DESCRIPTION:
                        016B   204 ;
                        016B   205 ;                    CONDX AND CONDX_CLEANUP ARE SUBROUTINES WHICH ARE EXECUTED
                        016B   206 ; BEFORE AND AFTER THE VERIFY SUBROUTINE, RESPECTIVELY, WHENEVER A NEW
                        016B   207 ; CONDITION X VALUE IS SELECTED (SEE FUNCTIONAL DESCRIPTION OF SUCCOMMON
                        016B   208 ; ROUTINE IN SUCCOMMON.MAR). ANY SETUP FUNCTION PARTICULAR TO THE
                        016B   209 ; CONDITION X TABLE IS INCLUDED IN THE CONDX SUBROUTINE AND CLEANED
                        016B   210 ; UP, IF NECESSARY, IN THE CONDX_CLEANUP SUBROUTINE. THIS INCLUDES,
                        016B   211 ; ESPECIALLY, CODE TO DETECT CONFLICTS AMONG CURRENT ENTRIES IN TWO
                        016B   212 ; OR MORE CONDITION TABLES. IF A CONFLICT IS DETECTED, A NON-ZERO
                        016B   213 ; VALUE IS STORED INTO CONFLICT, WHICH CAUSES THE CALLING ROUTINE
                        016B   214 ; (SUCCOMMON) TO SKIP THE CURRENT ENTRY IN THE CONDITION X TABLE.
                        016B   215 ;
                        016B   216 ; CALLING SEQUENCE:
                        016B   217 ;
                        016B   218 ;       BSBW CONDX   BSBW CONDX_CLEANUP
                        016B   219 ;         WHERE X = 1,2,3,4,5
                        016B   220 ;
                        016B   221 ; INPUT PARAMETERS:
                        016B   222 ;
                        016B   223 ;       CONFLICT = 0
                        016B   224 ;
                        016B   225 ; IMPLICIT INPUTS:
                        016B   226 ;
                        016B   227 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                        016B   228 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                        016B   229 ;
                        016B   230 ; OUTPUT PARAMETERS:
                        016B   231 ;
                        016B   232 ;       CONFLICT SET TO NON-ZERO IF COND TABLE CONFLICT DETECTED.
                        0162   233 ;
                        016B   234 ; IMPLICIT OUTPUTS:
                        016B   235 ;
                        016B   236 ;       R2,3,4,5,6 PRESERVED
                        016B   237 ;
                        016B   238 ; COMPLETION CODES:
                        016B   239 ;
                        016B   240 ;       NONE
                        016B   241 ;
                        016B   242 ; SIDE EFFECTS:
                        016B   243 ;
                        016B   244 ;       NONE
                        016B   245 ;
                        016B   246 ;--
                        016B   247
                        016B   248
                        016B   249
                        016B   250 COND1::
                  05    016B   251          RSB                             ; RETURN TO MAIN ROUTINE
                        016C   252 COND1_CLEANUP::
                  05    016C   253          RSB                             ; RETURN TO MAIN ROUTINE
                        016D   254 COND2::
                  05    016D   255          RSB                             ; RETURN TO MAIN ROUTINE
                        016E   256 COND2_CLEANUP::
                  05    016E   257          RSB                             ; RETURN TO MAIN ROUTINE
```

```
                016F    258 COND3::
 05             016F    259         RSB                             ; RETURN TO MAIN ROUTINE
                0170    260 COND3_CLEANUP::
 05             0170    261         RSB                             ; RETURN TO MAIN ROUTINE
                0171    262 COND4::
 05             0171    263         RSB                             ; RETURN TO MAIN ROUTINE
                0172    264 COND4_CLEANUP::
 05             0172    265         RSB                             ; RETURN TO MAIN ROUTINE
                0173    266 COND5::
 05             0173    267         RSB                             ; RETURN TO MAIN ROUTINE
                0174    268 COND5_CLEANUP::
 05             0174    269         RSB                             ; RETURN TO MAIN ROUTINE
```

```
                        0175   271          .SBTTL  FORM_CONDS
                        0175   272 ;++
                        0175   273 ; FUNCTIONAL DESCRIPTION:
                        0175   274 ;
                        0175   275 ;               FORM_CONDS FORMATS AND PRINTS INFORMATION ABOUT
                        0175   276 ;  THE CURRENT ELEMENT IN EACH OF THE CONDITION TABLES.
                        0175   277 ;
                        0175   278 ; CALLING SEQUENCE:
                        0175   279 ;
                        0175   280 ;       BSBW FORM_CONDS
                        0175   281 ;
                        0175   282 ; INPUT PARAMETERS:
                        0175   283 ;
                        0175   284 ;       NONE
                        0175   285 ;
                        0175   286 ; IMPLICIT INPUTS:
                        0175   287 ;
                        0175   288 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                        0175   289 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                        0175   290 ;       FOR X = 1,2,3,4,5 :
                        0175   291 ;               CONDX_T - TITLE TEXT FOR CONDX TABLE
                        0175   292 ;               CONDX_TAB - ELEMENT TEXT FOR CONDX TABLE
                        0175   293 ;               CONDX_C - CONTEXT OF THE CONDX TABLE
                        0175   294 ;               CONDX_E - DATA ELEMENTS OF THE CONDX TABLE
                        0175   295 ;
                        0175   296 ; OUTPUT PARAMETERS:
                        0175   297 ;
                        0175   298 ;       NONE
                        0175   299 ;
                        0175   300 ; IMPLICIT OUTPUTS:
                        0175   301 ;
                        0175   302 ;       NONE
                        0175   303 ;
                        0175   304 ; COMPLETION CODES:
                        0175   305 ;
                        0175   306 ;       NONE
                        0175   307 ;
                        0175   308 ; SIDE EFFECTS:
                        0175   309 ;
                        0175   310 ;       NONE
                        0175   311 ;
                        0175   312 ;--
                        0175   313
                        0175   314
                        0175   315
                        0175   316 FORM_CONDS::
                        0175   317          $FAO_S  MSG1_INP_CTL,FAO_LEN,FAO_DESC,TESTNUM
                        0194   318                                          ; FORMAT CONDITIONS HEADER MSG
                   FE69' 30     0194   319          BSBW    OUTPUT_MSG      ;  ... AND PRINT IT
                      14 00 91  0197   320          CMPB    #COND1_C,#NULL  ; IS CONDITION 1 NULL ?
                         03 12  019A   321          BNEQU   10$             ; NO -- CONTINUE
                      00BF 31   019C   322          BRW     FORM_CONDSX     ; YES -- SUBROUTINE IS FINISHED
                            019F   323 10$:
  00000000'EF   00000118'EF DE  019F   324          MOVAL   COND1_T,MSG_A   ; SAVE ADDRESS OF CONDITION 1 TITLE FOR FAO
00000000'EF   00000147'EF42 D0  01AA   325          MOVL    COND1_TAB[R2],MSG_B ; SAVE ADDR OF COND 1 CURR TEXT ELT FOR FAO
        00000000'EF    00 90    01B6   326          MOVB    #COND1_C,MSG_CTXT   ; SAVE CONDITION 1 CONTEXT FOR FAO
                               01BD   327          MOV_VAL COND1_C,COND1_E[R2],MSG_DATA1 ; GIVE COND 1 DATA VALUE TO FAO
```

```
                                                     B 1
                     FE40'   30   01BD   328          BSBW    WRITE_MSG2                      ; FORMAT AND WRITE CONDITION 1 MSG
                  14   00   91   01C0   329          CMPB    #COND2_C,#NULL                  ; IS CONDITION 2 NULL ?
                       03   12   01C3   330          BNEQU   20$                             ; NO -- CONTINUE
                     0096   31   01C5   331          BRW     FORM_CONDSX                     ; YES -- SUBROUTINE IS FINISHED
                            01C8   332 20$:
  00000000'EF   0000016B'EF   DE   01C8   333          MOVAL   COND2_T,MSG_A                   ; SAVE ADDRESS OF CONDITION 2 TITLE FOR FAO
00000000'EF   00000198'EF43   D0   01D3   334          MOVL    COND2_TAB[R3],MSG_B             ; SAVE ADDR OF COND 2 CURR TEXT ELT FOR FAO
        00000000'EF   00   90   01DF   335          MOVB    #COND2_C,MSG_CTXT               ; SAVE CONDITION 2 CONTEXT FOR FAO
                            01E6   336          MOV_VAL COND2_C,COND2_E[R3],MSG_DATA1 ; GIVE COND 2 DATA VALUE TO FAO
                     FE17'   30   01E6   337          BSBW    WRITE_MSG2                      ; FORMAT AND WRITE CONDITION 2 MSG
                  14   14   91   01E9   338          CMPB    #COND3_C,#NULL                  ; IS CONDITION 3 NULL ?
                       03   12   01EC   339          BNEQU   30$                             ; NO -- CONTINUE
                     006D   31   01EE   340          BRW     FORM_CONDSX                     ; YES -- SUBROUTINE IS FINISHED
                            01F1   341 30$:
  00000000'EF   00000217'EF   DE   01F1   342          MOVAL   COND3_T,MSG_A                   ; SAVE ADDRESS OF CONDITION 3 TITLE FOR FAO
00000000'EF   00000217'EF44   D0   01FC   343          MOVL    COND3_TAB[R4],MSG_B             ; SAVE ADDR OF COND 3 CURR TEXT ELT FOR FAO
        00000000'EF   14   90   0208   344          MOVB    #COND3_C,MSG_CTXT               ; SAVE CONDITION 3 CONTEXT FOR FAO
                            020F   345          MOV_VAL COND3_C,COND3_E[R4],MSG_DATA1 ; GIVE COND 3 DATA VALUE TO FAO
                     FDEE'   30   020F   346          BSBW    WRITE_MSG2                      ; FORMAT AND WRITE CONDITION 3 MSG
                  14   14   91   0212   347          CMPB    #COND4_C,#NULL                  ; IS CONDITION 4 NULL ?
                       47   13   0215   348          BEQLU   FORM_CONDSX                     ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   00000218'EF   DE   0217   349          MOVAL   COND4_T,MSG_A                   ; SAVE ADDRESS OF CONDITION 4 TITLE FOR FAO
00000000'EF   00000218'EF45   D0   0222   350          MOVL    COND4_TAB[R5],MSG_B             ; SAVE ADDR OF COND 4 CURR TEXT ELT FOR FAO
        00000000'EF   14   90   022E   351          MOVB    #COND4_C,MSG_CTXT               ; SAVE CONDITION 4 CONTEXT FOR FAO
                            0235   352          MOV_VAL COND4_C,COND4_E[R5],MSG_DATA1 ; GIVE COND 4 DATA VALUE TO FAO
                     FDC8'   30   0235   353          BSBW    WRITE_MSG2                      ; FORMAT AND WRITE CONDITION 4 MSG
                  14   14   91   0238   354          CMPB    #COND5_C,#NULL                  ; IS CONDITION 5 NULL ?
                       21   13   023B   355          BEQLU   FORM_CONDSX                     ; YES -- SUBROUTINE IS FINISHED
  00000000'EF   00000219'EF   DE   023D   356          MOVAL   COND5_T,MSG_A                   ; SAVE ADDRESS OF CONDITION 5 TITLE FOR FAO
00000000'EF   00000219'EF46   D0   0248   357          MOVL    COND5_TAB[R6],MSG_B             ; SAVE ADDR OF COND 5 CURR TEXT ELT FOR FAO
        00000000'EF   14   90   0254   358          MOVB    #COND5_C,MSG_CTXT               ; SAVE CONDITION 5 CONTEXT FOR FAO
                            025B   359          MOV_VAL COND5_C,COND5_E[R6],MSG_DATA1 ; GIVE COND 5 DATA VALUE TO FAO
                     FDA2'   30   025B   360          BSBW    WRITE_MSG2                      ; FORMAT AND WRITE CONDITION 5 MSG
                            025E   361 FORM_CONDSX:
                       05   025E   362          RSB                                     ; RETURN TO CALLER
```

```
                                .SBTTL  VERIFY
                        ;++
                        ; FUNCTIONAL DESCRIPTION:
                        ;
                        ;       VERIFY IS CALLED ONCE FOR EACH COMBINATION OF CONDITION
                        ; TABLE VALUES (AS DETERMINED BY THE INDEX REGISTERS R2,3,4,5,6 FOR
                        ; COND TABLES 1,2,3,4,5, RESPECTIVELY). VERIFY ESTABLISHES THE CONDITIONS
                        ; SPECIFIED BY THE COND TABLES AND ISSUES THE SUBJECT SYSTEM SERVICE
                        ; ($DLCEFC). THEN, THE SUCCESSFUL OPERATION OF THE SERVICE IS VERIFIED
                        ; BY EXAMINING THE STATUS CODE RETURNED, THE VALUES FOR RETURN ARGUMENTS
                        ; AND THE FUNCTIONALITY PERFORMED. THE EXAMINATIONS TAKE THE FORM OF
                        ; COMPARISONS AGAINST EXPECTED VALUES. ANY FAILING COMPARISON CAUSES AN
                        ; ERR_EXIT MACRO TO BE EXECUTED (EITHER DIRECTLY, OR INDIRECTLY,
                        ; THROUGH THE SS_CHECK MACRO); ERR_EXIT SETS EFLAG TO NON-ZERO,
                        ; PRINTS ERROR MESSAGES AND CAUSES AN IMMEDIATE RSB TO CALLER.
                        ; WHEN ERR_EXIT IS EXECUTED, FURTHER CALLS TO VERIFY ARE SUPPRESSED,
                        ; AND, AFTER EXECUTING CLEANUP SUBROUTINES, THE IMAGE EXITS.
                        ;
                        ; CALLING SEQUENCE:
                        ;
                        ;       BSBW VERIFY
                        ;
                        ; INPUT PARAMETERS:
                        ;
                        ;       NONE
                        ;
                        ; IMPLICIT INPUTS:
                        ;
                        ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
                        ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
                        ;       FOR X = 1,2,3,4,5 :
                        ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
                        ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
                        ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
                        ;                 FOR CONDX_E.
                        ;
                        ; OUTPUT PARAMETERS:
                        ;
                        ;       NONE
                        ;
                        ; IMPLICIT OUTPUTS:
                        ;
                        ;       VERIFY HAS NO OUTPUT. SINCE ITS PURPOSE IS TO TEST FOR ERRORS,
                        ;       IT MERELY RETURNS TO CALLER NORMALLY AFTER THE TESTS, PROVIDING
                        ;       ALL WERE SUCCESSFUL; IF AN ERROR IS DISCOVERED, RETURN IS VIA
                        ;       AN ERR_EXIT OR SS_CHECK MACRO, BOTH OF WHICH DOCUMENT DETECTED
                        ;       ERRORS.
                        ;
                        ; COMPLETION CODES:
                        ;
                        ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
                        ;
                        ; SIDE EFFECTS:
                        ;
                        ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
                        ;       (VIA RSB) IF ERROR ENCOUNTERED.
                        ;
```

```
                                                D 1
                     025F   421 ;--
                     025F   422
                     025F   423
                     025F   424
                     025F   425 VERIFY::
00000000'EF   95     025F   426         TSTB    CFLAG                   ; SHOULD CONDITIONS BE PRINTED ?
          03   13     0265   427         BEQL    5$                      ; NO -- CONTINUE
        FF0B   30     0267   428         BSBW    FORM_CONDS              ; YES -- FMT & PRINT ALL CONDS FOR THIS T.C.
                     026A   429 5$:
                     026A   430 ;
                     026A   431 ;  CREATE A CLUSTER AND GET IT BUILT AT LEAST ONCE, THEN DELETE
                     026A   432 ;    IT WITH A DACEFC.
                     026A   433 ;
                     026A   434         $ASCEFC_S EFN=#64, NAME=CLUS_NAME ; CREATE A NEW CLUSTER
                     0281   435         SS_CHECK NORMAL                 ; CHECK ITS COMPLETION
   5A   40 8F   9A   02AF   436         MOVZBL  #64,R10                 ; IDENTIFY EVENT FLAG GROUP TO BUILD_CLUST
        04EA   30     02B3   437         BSBW    BUILD_CLUST             ; BUILD THE SUBJECT CLUSTER
00000000'EF   95     02B6   438         TSTB    EFLAG                   ; IS AN ERROR BEING PROCESSED ?
          03   13     02BC   439         BEQL    10$                     ; NO -- CONTINUE
        04A2   31     02BE   440         BRW     VERIFYX                 ; YES -- RETURN IMMEDIATELY
                     02C1   441 10$:
                     02C1   442         $SETEF_S EFN=#95                ; ENSURE CLUSTER HAS AT LEAST ONE FLAG ON
                     02CE   443         SS_CHECK WASCLR                 ; FLAG 95 SHOULD HAVE BEEN CLEAR PREVIOUSLY
                     02FC   444         $DACEFC_S EFN=#64               ; NOW GET RID OF TEMPORARY CLUSTER
                     0309   445         SS_CHECK NORMAL                 ; ... CHECK ITS COMPLETION
                     0337   446 ;
                     0337   447 ;  THE FOLLOWING CASE INSTRUCTION AND SUBSEQUENT CODING
                     0337   448 ;    ISSUES AS MANY ASCEFC'S AS NECESSARY TO ACHIEVE THE
                     0337   449 ;    SPECIFIED REFERENCE COUNT FOR THIS TEST CASE.
                     0337   450 ;
00000114'EF   40 8F   9A   0337   451         MOVZBL  #64,EFN_REFCT1          ; ASSUME EVENT FLAG GROUP 2
      04   00   53   8F   033F   452         CASEB   R3,#0,#4                ; ISSU CORRECT AS^EFC'S PER COND 2 INDEX REG
                     0343   453 15$:                                        ; START OF CASE WORD DISPLACEMENTS
                000D' 0343   454         .WORD   20$-15$                 ; REF COUNT 0
                0010' 0345   455         .WORD   30$-15$                 ; REF COUNT 1, EVENT FLAG GROUP 2
                005D' 0347   456         .WORD   40$-15$                 ; REF COUNT 1, EVENT FLAG GROUP 3
                00B2' 0349   457         .WORD   50$-15$                 ; REF COUNT 2
                00B2' 034B   458         .WORD   50$-15$                 ; REF COUNT 4
              01A5   31     034D   459         BRW     55$                     ; BRANCH PAST CASE ROUTINES
                     0350   460 20$:
                     0350   461 ;
                     0350   462 ;  REF COUNT 0, NO ASCEFC'S TO BE ISSUED
                     0350   463 ;
              01B7   31     0350   464         BRW     60$                     ; GO ON TO ISSUE SUBJECT DLCEFC
                     0353   465 30$:
                     0353   466 ;
                     0353   467 ;  REF COUNT 1, EVENT FLAG GROUP 2
                     0353   468 ;
                     0353   469         $ASCEFC_S EFN=EFN_REFCT1, -      ; INCREMENT REF COUNT; EFN SET UP ABOVE
                     0353   470                   NAME=CLUS_NAME, -
                     0353   471                   PERM=CONDT_E[R2]
                     036F   472         SS_CHECK NORMAL                 ; CHECK FOR NORMAL COMPLETION
              0155   31     039D   473         BRW     55$                     ; GO BUILD THE CLUSTER JUST CREATED
                     03A0   474 40$:
                     03A0   475 ;
                     03A0   476 ;  REF COUNT 1, EVENT FLAG GROUP 3
                     03A0   477 ;
```

```
                                                E 1
00000114'EF   60 8F   9A  03A0   478          MOVZBL  #96,EFN_REFCT1                  ; EST EFN FOR E.F. GROUP 3
                      03A8   479          $ASCEFC_S EFN=EFN_REFCT1, -                ; INCREMENT REF COUNT
                      03A8   480                    NAME=CLUS_NAME, -
                      03A8   481                    PERM=CONDT_E[R2]
                      03C4   482          SS_CHECK NORMAL                            ; CHECK FOR NORMAL STATUS CODE
                0100  31     03F2   483          BRW     55$                        ; GO BUILD CLUSTER JUST CREATED
                      03F5   484 50$:
                      03F5   485 ;
                      03F5   486 ;  REF COUNT 2 OR 4, BOTH EVENT FLAG GROUPS IN THIS PROCESS
                      03F5   487 ;
                      03F5   488          $ASCEFC_S EFN=#64, -                       ; INCREMENT REF COUNT
                      03F5   489                    NAME=CLUS_NAME, -
                      03F5   490                    PERM=CONDT_E[R2]
                      0411   491          SS_CHECK NORMAL                            ; CHECK FOR NORMAL COMPLETION
                      043F   492          $ASCEFC_S EFN=#96, -                       ; INCREMENT REF COUNT
                      043F   493                    NAME=CLUS_NAME, -
                      043F   494                    PERM=CONDT_E[R2]
                      045B   495          SS_CHECK NORMAL                            ; CHECK FOR NORMAL COMPLETION
          04    53    D1   0489   496          CMPL    R3,#4                      ; FIFTH COND 2 ELEMENT (REF COUNT 4) ?
                67    12   048C   497          BNEQU   55$                        ; NO -- REF COUNT IS 2 -- NO MORE ASCEFC'S
                      048E   498                                                     ; YES -- CREATED PROC ISSUE 2 MORE ASCEFC'S
                      048E   499          $CREPRC_S PRCNAM=CREPRN, IMAGE=IMAGNAM, -
                      048E   500                    MBXUNT=MBXUNIT, QUOTA=QUOTALIST
                      04C0   501          SS_CHECK NORMAL                            ; CHECK CREPRC COMPLETION STATUS
                      04EE   502          $HIBER_S                                   ; SLEEP UNTIL CREATED PROCESS DOES ASCEFC'S
                      04F5   503 55$:
                      04F5   504 ;
                      04F5   505 ;  END OF CASE ROUTINES
                      04F5   506 ;
                      04F5   507 ;  AT THIS POINT, AT LEAST ONE ASCEFC HAS BEEN ISSUED; CALL
                      04F5   508 ;    BUILD_CLUST TO GET THE CREATED CLUSTER BUILT.
                      04F5   509 ;
      5A  00000114'EF   D0   04F5   510          MOVL    EFN_REFCT1,R10             ; ESTABLISH EFN FOR BUILD_CLUST
                02A1  30     04FC   511          BSBW    BUILD_CLUST                ; BUILD CLUSTER
          00000000'EF  95     04FF   512          TSTB    EFLAG                      ; IS AN ERROR BEING PROCESSED ?
                   03  13     0505   513          BEQL    60$                        ; NO -- CONTINUE
                0259  31     0507   514          BRW     VERIFYX                    ; YES -- RETURN IMMEDIATELY
                      050A   515 60$:
                      050A   516 ;
                      050A   517 ;  ****** SYSTEM SERVICE CALL WHICH IS THE SUBJECT OF THIS TEST CASE ******
                      050A   518 ;
                      050A   519          $DLCEFC_S NAME=CLUS_NAME                   ; CLEAR PERMANENT INDICATOR
      00000000'8F   50    D1   0517   520          CMPL    R0,#SS$_NORMAL             ; CODE RECEIVED = CODE EXPECTED ?
                    61    13   051E   521          BEQLU   70$                        ; YES -- CONTINUE
00000000'EF 00000000'8F   D0   0520   522          MOVL    #SS$_NORMAL,EXPV           ; LOAD UP EXPECTED AND ...
      00000000'EF   50    D0   052B   523          MOVL    R0,RECV                    ; ... RECEIVED VALUES, THEN EXIT
                      0532   524          ERR_EXIT LONG,<INCORRECT STATUS CODE RETURNED FROM DLCEFC>
                      0581   525 70$:
                      0581   526 ;
                      0581   527 ;  TO VERIFY THE SUBJECT DLCEFC, THE CLUSTER'S REFERENCE COUNT
                      0581   528 ;    WILL BE DROPPED TO 0 (BY ISSUING DACEFC'S) AND THEN AN
                      0581   529 ;    ADDITIONAL ASCEFC WILL BE ISSUED TO CREATE A NEW CLUSTER
                      0581   530 ;    WHICH WILL BE VERIFIED TO BE ALL 0'S.  BEFORE EACH DACEFC
                      0581   531 ;    IS ISSUED, A READEF OF THE CLUSTER WILL BE DONE TO ENSURE
                      0581   532 ;    THAT THE CLUSTER REMAINS EQUAL TO THE CLUSTER MASK; THIS
                      0581   533 ;    GUARANTEES THAT THE REFERENCE COUNT WAS CORRECT AND, IN
                      0581   534 ;    FACT, THAT THE CLUSTER IS NOT DELETED UNTIL THE COUNT
```

```
                     0581    535 ;    GOES TO 0.
                     0581    536 ;
                     0581    537 ;
                     0581    538 ;  THE FOLLOWING CASE STATEMENT AND SUBSEQUENT CODING
                     0581    539 ;    DECREMENTS THE REFERENCE COUNT BY ISSUING THE CORRECT
                     0581    540 ;    SEQUENCE OF DACEFC'S, BASED ON THE REFERENCE COUNT
                     0581    541 ;    FOR THIS TEST CASE.
                     0581    542 ;
        04 00 53 8F  0581    543             CASEB   R3,#0,#4                ; ISSU CORRECT DACEFC'S PER COND 2 INDEX REG
                     0585    544 75$:                                       ; CASE INSTRUCTION WORD DISPLACEMENTS
              000D'  0585    545             .WORD   80$-75$                 ; REF COUNT 0
              0010'  0587    546             .WORD   90$-75$                 ; REF COUNT 1, EVENT FLAG GROUP 2
              0010'  0589    547             .WORD   90$-75$                 ; REF COUNT 1, EVENT FLAG GROUP 3
              0028'  058B    548             .WORD   100$-75$                ; REF COUNT 2
              0028'  058D    549             .WORD   100$-75$                ; REF COUNT 4
          00DB   31  058F    550             BRW     130$                    ; BRANCH PAST CASE ROUTINES
                     0592    551 80$:
                     0592    552 ;
                     0592    553 ;  REF COUNT 0, NO DACEFC NECESSARY
                     0592    554 ;
          00D8   31  0592    555             BRW     130$                    ; GO ON TO CHECK CLUSTER FOR 0'S
                     0595    556 90$:
                     0595    557 ;
                     0595    558 ;  REF COUNT 1, ISSUE ONE DACEFC
                     0595    559 ;
     5A 00000114'EF  D0  0595    560             MOVL    EFN_REFCT1,R10          ; SET UP CORRECT EFN FOR READ_DACEFC SUBRTN
            02A7   30  059C    561             BSBW    READ_DACEFC             ; CHECK THE CLUSTER AND DISASSOCIATE
     00000000'EF   95  059F    562             TSTB    EFLAG                   ; IS AN ERROR BEING PROCESSED ?
              03   13  05A5    563             BEQL    95$                     ; NO -- CONTINUE
            01B9   31  05A7    564             BRW     VERIFYX                 ; YES -- RETURN IMMEDIATELY
                     05AA    565 95$:
            00C0   31  05AA    566             BRW     130$                    ; GO ON TO CHECK CLUSTER FOR 0'S
                     05AD    567 100$:
                     05AD    568 ;
                     05AD    569 ;  REF COUNT 2 OR 4, ISSUE 2 OR 4 DACEFC'S
                     05AD    570 ;
        04    53   D1  05AD    571             CMPL    R3,#4                   ; FIFTH COND 2 ELEMENT (REF COUNT 4) ?
              03   13  05B0    572             BEQLU   105$                    ; YES -- CONTINUE
            0094   31  05B2    573             BRW     110$                    ; NO -- REF COUNT MUST BE 2
                     05B5    574 105$:
                     05B5    575             $WAKE_S PRCNAM=CREPRN           ; WAKE PROCESS TO GET DACEFC'S ISSUED
                     05C4    576             SS_CHECK NORMAL                 ; CHECK FOR NORMAL STATUS CODE
                     05F2    577             $QIOW_S CHAN=MBXCHAN, FUNC=#IO$_READVBLK, -
                     05F2    578                     P1=MBXBUFF+8, P2=MBXBUFF
                     061B    579                                             ; ... AND WAIT FOR IT TO SEND MAIL
                     061B    580             SS_CHECK NORMAL                 ; CHECK FOR NORMAL STATUS CODE
                     0649    581 110$:
     5A    40 8F    9A  0649    582             MOVZBL  #64,R10                 ; SET UP CORRECT EFN FOR READ_DACEFC SUBRTN
            01F6   30  064D    583             BSBW    READ_DACEFC             ; CHECK THE CLUSTER AND DISASSOCIATE
 00000000'EF   95  0650    584             TSTB    EFLAG                   ; IS AN ERROR BEING PROCESSED ?
              03   13  0656    585             BEQL    120$                    ; NO -- CONTINUE
            0108   31  0658    586             BRW     VERIFYX                 ; YES -- RETURN IMMEDIATELY
                     065B    587 120$:
     5A    60 8F    9A  065B    588             MOVZBL  #96,R10                 ; SET UP CORRECT EFN FOR READ_DACEFC SUBRTN
            01E4   30  065F    589             BSBW    READ_DACEFC             ; CHECK THE CLUSTER AND DISASSOCIATE
 00000000'EF   95  0662    590             TSTB    EFLAG                   ; IS AN ERROR BEING PROCESSED ?
              03   13  0668    591             BEQL    130$                    ; NO -- CONTINUE
```

```
                     00F6   31   066A   592                 BRW     VERIFYX                 ; YES -- RETURN IMMEDIATELY
                            066D   593 130$:
                            066D   594 ;
                            066D   595 ;  REFERENCE COUNT SHOULD BE DOWN TO 0 AND CLUSTER DELETED.
                            066D   596 ;    DO ONE MORE ASCEFC AND EXPECT A CLUSTER OF 0'S.
                            066D   597 ;
                            066D   598                 $ASCEFC_S EFN=#64, NAME=CLUS_NAME ; RE-ASSOCIATE SAME CLUSTER
                            0684   599                 SS_CHECK NORMAL                   ; CHECK COMPLETION STATUS
                            06B2   600                 $READEF_S EFN=#64, STATE=CLUS_STATE ; READ CLUSTER
                         2E 50   E8 06C5   601                 BLBS    R0,140$                   ; CONTINUE IF NORMAL COMPLETION
                            06C8   602                 SS_CHECK NORMAL                   ; USE SS_CHECK TO TERMINATE TEST MODULE
                            06F6   603 140$:
                 00000110'EF   D5 06F6   604                 TSTL    CLUS_STATE                ; IS CLUSTER INITIALIZED TO 0'S ?
                           65   13 06FC   605                 BEQLU   VERIFYX                   ; YES -- THIS TEST CASE COMPLETE
                 00000000'EF   D4 06FE   606                 CLRL    EXPV                      ; NO -- LOAD EXPECTED AND ...
     00000000'EF 00000110'EF   D0 0704   607                 MOVL    CLUS_STATE,RECV           ; ... RECEIVED VALUES, THEN EXIT
                            070F   608                 ERR_EXIT LONG,<DELETED AND RE-ASSOCIATED CLUSTER NOT RE-INIT'D>
                            0763   609 VERIFYX:
                            05 0763   610                 RSB                               ; RETURN TO CALLER
```

```
                                    H 1
0764   612          .SBTTL  VFY_CLEANUP
0764   613 ;++
0764   614 ; FUNCTIONAL DESCRIPTION:
0764   615 ;
0764   616 ;               VFY_CLEANUP EXECUTES SYSTEM SERVICES TO UNDO THE
0764   617 ; EFFECT OF THOSE ISSUED IN THE VERIFY SUBROUTINE. VFY_CLEANUP MUST
0764   618 ; ASSUME THAT VERIFY MAY NOT HAVE EXECUTED IN ITS ENTIRETY (IF AN
0764   619 ; ERROR IS FOUND). ALSO, VFY_CLEANUP MAY ISSUE SS_CHECK OR ERR_EXIT
0764   620 ; ONLY AFTER PERFORMING ALL OF ITS CLEANUP OPERATIONS; THIS IS REQUIRED
0764   621 ; IN THE EVENT THAT VFY_CLEANUP IS CALLED DURING ERROR PROCESSING,
0764   622 ; WHEN PERFORMING THE REQUIRED CLEANUP IS MORE IMPORTANT THAN
0764   623 ; POSSIBLY DISCOVERING A SECOND ERROR.
0764   624 ;
0764   625 ; CALLING SEQUENCE:
0764   626 ;
0764   627 ;       BSBW VFY_CLEANUP
0764   628 ;
0764   629 ; INPUT PARAMETERS:
0764   630 ;
0764   631 ;       NONE
0764   632 ;
0764   633 ; IMPLICIT INPUTS:
0764   634 ;
0764   635 ;       R2,3,4,5,6 CONTAIN CURRENT CONDITION TABLE INDEX VALUES
0764   636 ;         FOR COND TABLES 1,2,3,4,5, RESPECTIVELY.
0764   637 ;       FOR X = 1,2,3,4,5 :
0764   638 ;               CONDX_E - ADDRESS OF TABLE OF DATA VALUES FOR CONDX
0764   639 ;                 TABLE. IF THE CONTEXT OF TABLE X IS A SYSTEM SERVICE
0764   640 ;                 ARGUMENT, THE ARGUMENT NAME MAY BE USED AS A SYNONYM
0764   641 ;                 FOR CONDX_E.
0764   642 ;
0764   643 ; OUTPUT PARAMETERS:
0764   644 ;
0764   645 ;       NONE
0764   646 ;
0764   647 ; IMPLICIT OUTPUTS:
0764   648 ;
0764   649 ;       NONE
0764   650 ;
0764   651 ; COMPLETION CODES:
0764   652 ;
0764   653 ;       EFLAG SET TO NON-ZERO IF ERROR ENCOUNTERED.
0764   654 ;
0764   655 ; SIDE EFFECTS:
0764   656 ;
0764   657 ;       SS_CHECK AND ERR_EXIT MACROS CAUSE PREMATURE EXIT
0764   658 ;       (VIA RSB) IF ERROR ENCOUNTERED.
0764   659 ;
0764   660 ;--
0764   661
0764   662
0764   663
0764   664 VFY_CLEANUP::
0764   665         $DLCEFC_S NAME=CLUS_NAME        ; CLEAR PERM INDICATOR IF PRESENT ...
0771   666         SS_CHECK NORMAL                 ; ... AND CHECK COMPLETION
05 079F   667         RSB                             ; RETURN TO CALLER
```

```
                                                07A0    669          .SBTTL  BUILD_CLUST SUBROUTINE
                                                07A0    670 ;
                                                07A0    671 ;*****************************************************************
                                                07A0    672 ;*
                                                07A0    673 ;*     BUILD_CLUST SUBROUTINE
                                                07A0    674 ;*
                                                07A0    675 ;*     THIS SUBROUTINE CREATES A 32-BIT CLUSTER MASK BY
                                                07A0    676 ;*       CONCATENATING THE LOW-ORDER BYTES OF REGS R2-R5.
                                                07A0    677 ;*       IT THEN SETS CLUSTER A EQUAL TO THE MASK BY
                                                07A0    678 ;*       ISSUING THE PROPER COMBINATION OF 32 SETEF/CLREF'S.
                                                07A0    679 ;*
                                                07A0    680 ;*     INPUTS:
                                                07A0    681 ;*
                                                07A0    682 ;*             R2,R3,R4,R5   -  CONDITION TABLE INDEX VALUES
                                                07A0    683 ;*
                                                07A0    684 ;*             R10      -       ANY EFN IN CLUSTER A
                                                07A0    685 ;*
                                                07A0    686 ;*     OUTPUTS:
                                                07A0    687 ;*
                                                07A0    688 ;*             CLUS_MASK   -    LONGWORD CONTAINING THE CREATED
                                                07A0    689 ;*                                CLUSTER MASK.
                                                07A0    690 ;*
                                                07A0    691 ;*             SUBJECT CLUSTER - UPDATED TO LOOK LIKE CLUS_MASK.
                                                07A0    692 ;*
                                                07A0    693 ;*     VOLATILE REGISTERS:
                                                07A0    694 ;*
                                                07A0    695 ;*             R0, R1, R8, R9
                                                07A0    696 ;*
                                                07A0    697 ;*****************************************************************
                                                07A0    698 ;
                                                07A0    699 BUILD_CLUST:
0000010C'EF    55    90                         07A0    700          MOVB    R5,CLUS_MASK            ; BUILD
0000010D'EF    54    90                         07A7    701          MOVB    R4,CLUS_MASK+1          ; .. CLUSTER
0000010E'EF    53    90                         07AE    702          MOVB    R3,CLUS_MASK+2          ; ..... MASK
0000010F'EF    52    90                         07B5    703          MOVB    R2,CLUS_MASK+3          ; ..............
                                                07BC    704 ;
                                                07BC    705 ; THE FOLLOWING CODE SETS SUBJECT CLUSTER EQUAL TO CLUS_MASK
                                                07BC    706 ;
          58    5A    D0                        07BC    707          MOVL    R10,R8                  ; ESTABLISH FIRST EFN (EVENT FLAG NO.)
                59    D4                        07BF    708          CLRL    R9                      ; INIT OFFSET INTO CLUS_MASK
                                                07C1    709 20$:
3A 0000010C'EF  59    E0                        07C1    710          BBS     R9,CLUS_MASK,30$        ; ISSUE $SETEF IF BIT FOR THIS FLAG IS SET
                                                07C9    711          $CLREF_S EFN=R8                 ; ... OTHERWISE, ISSUE $CLREF
            68 50     E8                        07D2    712          BLBS    R0,40$                  ; IF NORMAL STATUS, PROCESS NEXT EVENT FLAG
                                                07D5    713          SS_CHECK NORMAL                 ; USE SS_CHECK TO TERMINATE TEST MODULE
                                                0803    714 30$:
                                                0803    715          $SETEF_S EFN=R8                 ; SET CURRENT EVENT FLAG
            2E 50     E8                        080C    716          BLBS    R0,40$                  ; IF NORMAL STATUS, PROCESS NEXT EVENT FLAG
                                                080F    717          SS_CHECK NORMAL                 ; USE SS_CHECK TO TERMINATE TEST MODULE
                                                083D    718 40$:
          58    B6                              083D    719          INCW    R8                      ; GET NEXT EFN
FF7C 59   01    1F                              083F    720          ACBB    #31,#1,R9,20$           ; GO DO NEXT EVENT FLAG
                05                              0845    721          RSB                             ; RETURN TO CALLER
```

```
                                        J 1
                                        .SBTTL  READ_DACEFC SUBROUTINE
                                ;
                                ;*****************************************************************
                                ;*
                                ;*      READ_DACEFC SUBROUTINE
                                ;*
                                ;*      THIS SUBROUTINE COMPARES THE CLUSTER AGAINST THE MASK
                                ;*        AND THEN DISASSOCIATES THE EVENT FLAG GROUP SPECIFIED
                                ;*        BY R10 FROM THE CLUSTER.
                                ;*
                                ;*      INPUTS:
                                ;*
                                ;*              R10     -       ANY EFN IN THE EVENT FLAG GROUP
                                ;*                                TO BE DISASSOCIATED.
                                ;*
                                ;*              CLUS_MASK   -   LONGWORD CONTAINING THE IMAGE
                                ;*                                OF THE SUBJECT CLUSTER.
                                ;*      OUTPUTS:
                                ;*
                                ;*              NONE.
                                ;*
                                ;*      EXITS:
                                ;*
                                ;*              NORMAL      -   RSB TO CALLER IF CLUSTER EQUALS
                                ;*                                MASK AND SYSTEM SERVICES
                                ;*                                FINISH NORMALLY.
                                ;*
                                ;*              ERROR       -   ERR_EXIT MACRO EXECUTED IF ABOVE
                                ;*                                NORMAL CONDITIONS ARE NOT MET.
                                ;*
                                ;*      VOLATILE REGISTERS, AREAS:
                                ;*
                                ;*              R0, R1, R8, R9, CLUS_STATE, EXPV, RECV
                                ;*
                                ;*****************************************************************
                                ;
                                READ_DACEFC:
                                        $READEF_S EFN=R10, STATE=CLUS_STATE ; READ CLUSTER
                        2E 50   E8      BLBS    R0,10$                  ; CONTINUE IF NORMAL COMPLETION
                                        SS_CHECK NORMAL                 ; USE SS_CHECK TO TERMINATE TEST MODULE
                                10$:
0000010C'EF     00000110'EF     D1      CMPL    CLUS_STATE,CLUS_MASK    ; DOES CLUSTER STILL = MASK ?
                        6F      13      BEQLU   20$                     ; YES -- ISSUE DACEFC AND GET OUT
00000000'EF     0000010C'EF     D0      MOVL    CLUS_MASK,EXPV          ; NO -- LOAD EXPECTED AND ...
00000000'EF     00000110'EF     D0      MOVL    CLUS_STATE,RECV         ; ... RECEIVED VALUES, THEN EXIT
                                        ERR_EXIT LONG,<CLUSTER STATE NOT MAINTAINED ACROSS DLCEFC OR DACEFC>
                                20$:
                                        $DACEFC_S EFN=R10               ; DISASSOCIATE (DECR REF CT BY 1)
                                        SS_CHECK NORMAL                 ; CHECK FOR NORMAL COMPLETION
                                READ_DACEFCX:
                        05              RSB                             ; RETURN TO CALLER
                                        .END
```

```
                                                K 1
$$$$                            = 000008B3 R      04
$$$CHARS                        = 00000034
$$$CHARS1                       = 00000004
$$$CHARS2                       = 00000012
$$$CHARS3                       = 00000012
$$$CHARS4                       = 00000016
$$$CHARS5                       = 00000028
$$$COND_A                       = 00000004
$$$STRINGS                      = 00000001
$$$STRINGS2                     = 00000005
$$T1                            = 00000000
$$T2                            = 00000004
BUILD_CLUST                       000007A0 R      04
BYTE                            = 00000001   G
CFLAG                             ********     X  04
CHMRTN                            ********     X  04
CHM_CONT                          ********     X  04
CLUS_MASK                         0000010C R      03
CLUS_NAME                         00000065 R      02
CLUS_STATE                        00000110 R      03
COMP_SC                           ********     X  04
CONDT                             00000168 RG     04
COND1_C                         = 00000000
COND1_CLEANUP                     0000016C RG     04
COND1_E                           00000163 R      03
COND1_H                           00000146 RG     03
COND1_T                           00000118 R      03
COND1_TAB                         00000147 R      03
COND2                             0000016D RG     04
COND2_C                         = 00000000
COND2_CLEANUP                     0000016E RG     04
COND2_E                           00000217 R      03
COND2_H                           00000197 RG     03
COND2_T                           0000016B R      03
COND2_TAB                         00000198 R      03
COND3                             0000016F RG     04
COND3_C                         = 00000014
COND3_CLEANUP                     00000170 RG     04
COND3_H                           00000217 RG     03
COND3_T                           00000217 R      03
COND3_TAB                         00000217 R      03
COND4                             00000171 RG     04
COND4_C                         = 00000014
COND4_CLEANUP                     00000172 RG     04
COND4_H                           00000218 RG     03
COND4_T                           00000218 R      03
COND4_TAB                         00000218 R      03
COND5                             00000173 RG     04
COND5_C                         = 00000014
COND5_CLEANUP                     00000174 RG     04
COND5_H                           00000219 RG     03
COND5_T                           00000219 R      03
COND5_TAB                         00000219 R      03
CREPRN                            00000051 R      02
CTL$GL_PHD                        ********   X    04
DESC                            = 00000010   G
DIB$K_LENGTH                    = 00000074

DIB$W_UNIT                      = 0000000C
EFLAG                             ********     X  04
EFN_REFCT1                        00000114 R      03
EXPD                              ********     X  04
FAO_DESC                          ********     X  04
FAO_LEN                           ********     X  04
FORM_CONDS                        00000175 RG     04
FORM_CONDSX                       0000025E R      04
IMAGNAM                           0000007A R      02
IO$_READVBLK                      ********     X  04
LONG                            = 00000004   G
MBXBUFF                           0000008C R      03
MBXCHAN                           00000008 R      03
MBXCHANINFO                       0000000C R      03
MBXUNIT                           00000088 R      03
MOD_MSG_CODE                      ********     X  04
MOD_MSG_PRINT                     ********     X  04
MSG1_INP_CTL                      00000019 R      02
MSG3_ERR_CTL                      00000039 RG     02
MSG_A                             ********     X  04
MSG_B                             ********     X  04
MSG_CTXT                          ********     X  04
NOTARG                          = 00000000   G
NULL                            = 00000014   G
OUTPUT_MSG                        ********     X  04
PCV                               ********     X  04
PHD$Q_PRIVMSK                   = 00000000
PQL$_BYTLM                      = 00000003
PQL$_CPULM                      = 00000004
PQL$_FILLM                      = 00000006
PQL$_LISTEND                    = 00000000
PQL$_PGFLQUOTA                  = 00000007
PQL$_PRCLM                      = 00000008
PQL$_TQELM                      = 00000009
PRIVMASK                          00000000 R      03
PRIV_ARGS                       = 00000002
PROCESS_ERR                       ********   X    04
QUAD                            = 00000008   G
QUOTALIST                         00000099 R      02
READ_DACEFC                       00000846 R      04
READ_DACEFCX                      00000939 R      04
RECV                              ********     X  04
REST_REGS                         ********     X  04
SAVE_REGS                         ********     X  04
SS$_NORMAL                        ********     X  04
SS$_WASCLR                        ********     X  04
SUCCESS                           ********     X  04
SYS$ASCEFC                        ********    GX  04
SYS$CLREF                         ********    GX  04
SYS$CMKRNL                        ********    GX  04
SYS$CREMBX                        ********    GX  04
SYS$CREPRC                        ********    GX  04
SYS$DACEFC                        ********    GX  04
SYS$DELMBX                        ********    GX  04
SYS$DLCEFC                        ********    GX  04
SYS$FAO                           ********     X  04
SYS$GETCHN                        ********    GX  04
```

```
SYS$HIBER                       ********  GX   04
SYS$QIOW                        ********  GX   04
SYS$READEF                      ********  GX   04
SYS$SETEF                       ********  GX   04
SYS$SETPRN                      ********  GX   04
SYS$SETPRV                      ********  GX   04
SYS$WAKE                        ********  GX   04
TESTNUM                         ********   X   04
TEST_MOD_NAME                   00000000 RG    02
TEST_MOD_NAME_D                 00000009 R     02
TEST_MOD_SUCC                   ********   X   04
TMD_ADDR                        ********   X   04
TM_CLEANUP                      00000159 RG    04
TM_SETUP                        00000000 RG    04
VERIFY                          0000025F RG    04
VERIFYX                         00000763 R     04
VFY_CLEANUP                     00000764 RG    04
WORD                          = 00000002  G
WRITE_MSG2                      ********   X   04
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RODATA	000000BC (188.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC LONG
RWDATA	0000021A (538.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
SATSSS52	0000093A (2362.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	35	00:00:00.07	00:00:00.31
Command processing	140	00:00:00.68	00:00:02.93
Pass 1	292	00:00:08.83	00:00:16.59
Symbol table sort	0	00:00:00.60	00:00:00.63
Pass 2	158	00:00:02.39	00:00:03.13
Symbol table output	16	00:00:00.10	00:00:00.13
Psect synopsis output	2	00:00:00.03	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	645	00:00:12.70	00:00:23.76

The working set limit was 1650 pages.
47530 bytes (93 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 357 non-local and 80 local symbols.
774 source lines were read in Pass 1, producing 28 object records in Pass 2.
50 pages of virtual memory were used to define 40 macros.

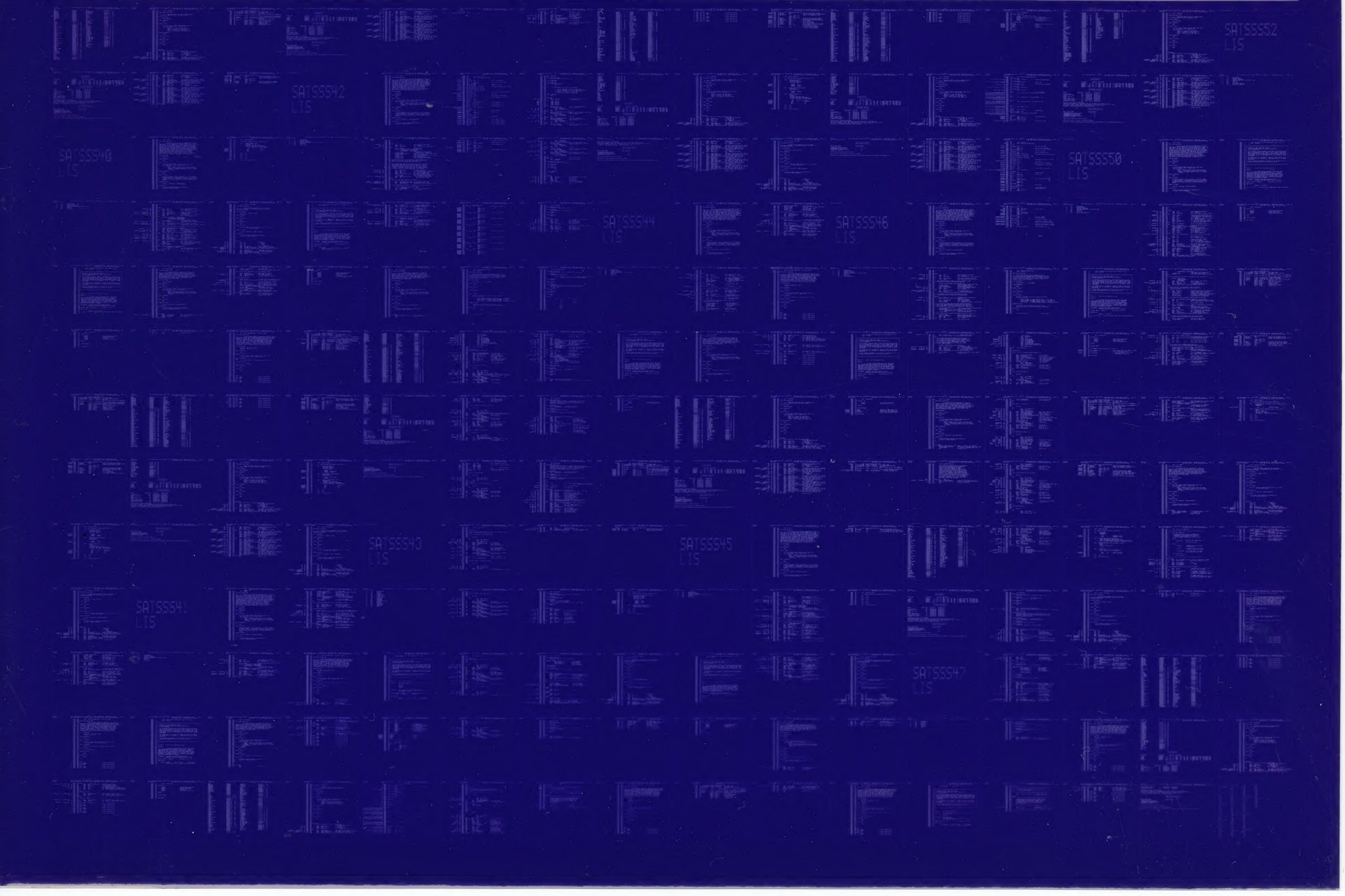
```
                              +----------------------------+
                              ! Macro library statistics !
                              +----------------------------+

Macro library name                     Macros defined
------------------                     --------------
_$255$DUA28:[SHRLIB]UETP.MLB;1                9
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                1
_$255$DUA28:[SYSLIB]STARLET.MLB;2            27
TOTALS (all libraries)                       37

775 GETS were required to define 37 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:SATSSS52/OBJ=OBJ$:SATSSS52 MSRC$:SATSSS52/UPDATE=(ENH$:SATSSS52)+EXECML$/LIB+SHRLIB$:UETP/LIB
```



0424 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SATSS556
LIS

SATSS571
LIS

SATSS560
LIS

SATSS554
LIS

SATSS573
LIS

SATSS570
LIS

SATSS572
LIS

SATSS555
LIS

SATSS553
LIS

SATSS561
LIS